Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/IL04/001073

International filing date: 22 November 2004 (22.11.2004)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/526,446
Filing date: 03 December 2003 (03.12.2003)

Date of receipt at the International Bureau: 03 January 2005 (03.01.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

IL04/1073

PA 1240875

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

November 01, 2004

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: ***60/526,446***
FILING DATE: ***December 03, 2003***

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

T. Lawrence

T. LAWRENCE
Certifying Officer

PTO/SB/16 (10-01)
Approved for use through10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Express Mail Label No.	ER 526816832 US

INVENTOR(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Gil	SEVER	76 Revivim St. Rosh-Haayin, 48621 Israel
Zvi	GUTTERMAN	24 Mara St. Jerusalem, 93715 Israel

☐ *Additional inventors are being named on the _____ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (500 characters max)

METHOD AND SYSTEM FOR IMPROVING COMPUTER NETWORK SECURITY

7548 U.S. PTO
60/526446
120303

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

☑ Customer Number 35856 → *Place Customer Number Bar Code Label here*

Type Customer Number here

OR

☐ Firm or Individual Name					
Address					
Address					
City		State		ZIP	
Country		Telephone		Fax	

ENCLOSED APPLICATION PARTS *(check all that apply)*

☑ Specification *Number of Pages* 30
☑ Drawing(s) *Number of Sheets* 5
☐ Application Data Sheet. See 37 CFR 1.76
☐ CD(s), Number
☐ Other (specify)

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT

☑ Applicant claims small entity status. See 37 CFR 1.27.
☐ A check or money order is enclosed to cover the filing fees
☐ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number:
☑ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($) $80.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☑ No.
☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

Respectfully submitted,

SIGNATURE _____

TYPED or PRINTED NAME Gregory Scott Smith

TELEPHONE 770.804.9070

Date 12/03/22003

REGISTRATION NO. *(if appropriate)* 40,819

Docket Number: 19013.0010

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

13146 U.S.PTO

PTO/SB/17 (10-03)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☑ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($) 80.00

Complete if Known	
Application Number	
Filing Date	Dec 3, 2003
First Named Inventor	SEVER et al.
Examiner Name	
Art Unit	
Attorney Docket No.	19013.0010

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☑ Credit card ☐ Money Order ☐ Other ☐ None

☐ Deposit Account:

Deposit Account Number
Deposit Account Name

The Director is authorized to: *(check all that apply)*

☑ Charge fee(s) indicated below ☑ Credit any overpayments

☐ Charge any additional fee(s) or any underpayment of fee(s)

☐ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	80.00

SUBTOTAL (1) ($) 80.00

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

	Extra Claims		Fee from below		Fee Paid
Total Claims		-20** =		X	=
Independent Claims		- 3** =		X	=
Multiple Dependent					=

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 0

SUBMITTED BY

(Complete *(if applicable)*)

Name (Print/Type)	Gregory S. Smith	Registration No. (Attorney/Agent)	40,819	Telephone	770.804.9070
Signature				Date	Dec 3, 2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

US PROVISIONAL PATENT APPLICATION.

TITLE:

METHOD AND SYSTEM FOR IMPROVING COMPUTER NETWORK SECURITY.

THE INVENTORS

Gil Sever, 76 Revivim St. Rosh-Haayin, 48621 Israel.

Zvi Gutterman, 24 Mara St. Jerusalem, 93715 Israel.

BACKGROUND

1. **Field of Invention:**

[0001] The present invention relates to the field of security of private network and more particularly to method for protecting private network from leakage or extraction of information or insertion of un approved material when the clients are connected to the private network or not connected (working online or offline).

2. **Description of Background Art**.

[0002] Commercial Corporations, enterprises, organization such government, military, financial etc. face several computer security concerns one of them is leakage of information from their internal computer network to the outside world. The threat of leakage of information may come from outsiders as well as from inside the organization by disloyal or careless employees.

[0003] Internal employees may use their permission to access enterprise's information, download it to their client computer and from there the information may be transferred to an external device. The external device may be a removable storage device (e.g. flash memory such as but not limited to DiscOnKey, removable hard disk drive), removable storage media (e.g., floppy disk, writable CD ROM), a PDA, a cellular

phone, WiFi dongle, Bluetooth dongle, etc. DiskOnKey is a registered trademark of M-Systems. A PDA is Short for personal digital assistant, a handheld device that may have computing, telephone/fax, Internet and networking features. Communication with those external devices may be done over variety of data communication ports such as USB, FireWire, PCMCIA bus, SCSI bus, iSCSI, Cellular, Infiniband, Serial, Parallel, LAN port, Fiber Channel, Infrared, wireless communication such as but not limited WiFi, Bluetooth, etc.

[0004] Another device that may be used for transferring information out of the organization is the employee's portable computer (e.g. a laptop computer). Today, in many organizations, a client may have a laptop computer instead of or in parallel to his desktop computer. An employee may copy to the laptop valuable information and carry it out. Later on, when the laptop is not connected to the private network the information may be copied to a storage device.

[0005] Common approach to deal with this type of threat is by preventing the access to those devices. Preventing the access may be done physically or by software means. However this common method reduces the productivity of the organization, since the user may need to access those devices during his day-to-day operation.

[0006] Therefore, there is a need for new method that may offer wider variety of option for controlling the information transfer and the access to those devices. A method that may permit transferring certain files but prevent others or may allow using some of the function of the external device and prevent others. For example, the method may allow synchronization of the diary in the user PC with the diary in his PDA but may prevent file transfer to the PDA etc.

[0007] Furthermore, there is a need for a method that may verify the environmental to which a portable device is connected. Check the option that

may be allowed in this location and reach a decision how to proceed. The decision may be based on a security policy that is loaded into the portable device. In addition, there is a need for a method that may analyze nesting of communication protocol in other communication protocol.

SUMMARY OF THE INVENTION

[0008] The present invention solves the above-described needs by providing a method for selectively preventing access to certain devices according to a security policy that is used. The method may analyze the data transportation to or from a communication port according to the relevant layers that are used in the communication protocol. Based on a policy and the type of the communication that has been analyzed, a decision is made whether to allow the transportation of the data, to block it, to inform the user and/or the administrator etc. The policy may be set by the administrator of the private network according to the user rights and position in the organization.

[0009] An exemplary embodiment of the present invention may have Security Manager Module and a plurality of client agents. The Security Manager Module (SMM) may reside on a security server in a central location in the private network and may manage the security policy. The SMM is operated by an administrator. The client agent, which may be a software and/or hardware, is installed in each one of the computers that can be connected to the private network. The existence of the agent is a mandatory condition for enabling the connection to the private network. The private network may not respond to a computer that does not have an agent.

[0010] The agent may sniff the data transportation to or from one or more of the communication ports or buses, analyzes it according to the communication protocol and may reach a decision how to proceed with the data transfer. The

policy that is associated with a certain agent may configure the agent to block the transportation of certain type of files such as but not limited to software code, source code, drawings etc. Or may allow certain application and block others. Moreover, the agent may be configured to send indication to the SMM and can be configured to send messages also to the user. The policy that is associated with each one of the clients may be updated from time to time by the administrator via the SMM.

[0011] For a communication that operates according to the Seven Layer Model the present invention may analyzes one or more layers from the existing layers in order to reach a decision. Moreover in cases in which the communication is using nesting of one protocol into under protocol, the present invention may analyze the one or more protocol and may reach a decision on the nested protocol. For example if a WiFi Dongle is connected to a USB port both protocols may be analyzed. A decision whether to allow the communication or not may be depended, for example, on the SSID property of the WiFi connection. The SSID property defines the name of the wireless network.

[0012] Moreover the present invention may verify if an external device that is connected to a communication port behaves as it is expected or if the device is emulate or impersonate another device. For example a portable memory that is connected over a USB port may be built to emulate a digital camera while establishing the connection. Later on the user may load into it some files. The present invention may detect such an activity and may block the communication.

[0013] Other objects, features, and advantages of the present invention will become apparent upon reading the following detailed description of the embodiments with the accompanying drawings and appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a block diagram of a computer system that uses an exemplary embodiment of the present invention:

FIG. 2 is a block diagram of a software system that is used in a client computer according to an exemplary embodiment of the present invention;

FIG. 3 is a block diagram illustrating components of the client security agent according to an exemplary embodiment of the present invention;

FIG. 4 illustrates a flowchart with the relevant steps of an exemplary method for managing input portion of data transportation via the security agent.

FIG. 5 illustrates a flowchart with the relevant steps of an exemplary method for determining how to proceed with a session of data transportation.

DETAILED DESCRIPTION OF THE INVENTION:

[0014] Turning now to the figures in which like numerals represent like elements throughout the several views, exemplary embodiments of the present invention are described. For convenience, only some elements of the same group may be labeled with numerals. The purpose of the drawings is to describe exemplary embodiments and not for production. Therefore features shown in the figures are chosen for convenience and clarity of presentation only.

[0015] FIG. 1 is a block diagram with the relevant elements of a computer system **100** that uses an exemplary embodiment of the present invention for protecting the computer system from their own clients. The computer system **100** may comprise a plurality of client computers **110a-c**, a private network **120**, a plurality of communication channels **115a-c** between the private network **120** and the plurality of client computers **110a-c**, and security server **130**. Three examples of user equipment **110a-c** and communication channels **115a-c** are shown in Fig. 1 by way of example, and any number other than three may be used with the present invention. The private network **120** may be an Intranet, cellular network, a LAN, a VPN (Virtual Private Network) or any other type of communication network.

[0016] The client computer **110a-c** may be a personal computer, a workstation, a desktop computer, mainframe computer, blade server (e.g. CITRIX), dumb terminal etc. or another type of computing device that can be connected over private network **120**. The client computer **110a-c** may also be a portable device, such as but not limited to a laptop computer, notebook computer, a personal digital assistant (PDA), or another type of mobile device. The client computer **110a-c** may connect to various networks from time to time at home, at work, and at other locations. Communication channels **115a-c**

may be permanent connection, temporary connection, and wire or wireless connection. A wireless connection may be RF connection that may be based on protocol like but not limited to Bluetooth or WiFi. The wireless connection may also be an Infra Red (IR) connection. More information about the client computer **110a-c** is disclosed below in conjunction with FIG. 2 and 3.

[0017] Security server **130** may be an element of network **120**. Security server **130** may be responsible to manage the security polices that are used over the private network **120**. A plurality of polices may be used by computer system **100**. The policies may be based on the client's degree of security, the environment that the client is working in, the type of the devices that are connected to the client computer, etc. The policies may be updated from time to time and may be loaded to the clients.

[0018] Security server **130** may ensure that all clients comply with specified security policies. For example, if a trigger event occurs and a copy of a corporate security policy is not available on a client computer **110a-c**, the client computer **110a-c** may initiate a connection to Security server **130**. In response the appropriate policies may be downloaded to the client computer. Security server **130** may update from time to time the security policies that are installed in each one of the client computers.

[0019] Security server **130** may comprise the following relevant modules: client's communication module **132**, event logger module **134**, policies database **135**, client database **136**, Man Machine Interface (MMI) module **139** and a manager module **138**. Communication module **132** may be used to communicate with the plurality of client computers **110a-c** over network **120** while the client computers are connected to the network **120**. The communication with the client may be encrypted to create a secure connection between the client and the security server **130**, over which data can be sent securely.

[0020] The communication from the security server **130** to the client **110** may include: updated security policies, periodically checking whether the installed security agent and the installed policies have been contaminated or have been tampered by any hostile code. If a particular client computer does not have a required client security agent or policy installed, or the agent was infected, the Security server **130** can cause access to the corporate network to be denied until such client has installed and activated the required agent or policy. The communication from the client to the server may include: on line indication when the device is connected to the network **120**, reports on events according to the security policy, reports on trials to affect the security agent or the stored security policy, etc. The report may include information on any connection of the client computer to an external devices, information on the data transfer, the timing of the event, the location, the device to which the data transfer was done, shadowing of the information that was transferred while the client computer was not connected or connected to network **120**, etc.

[0021] The event logger **134** may be a storage device that may be used to store the reports that have been sent from the users along a certain period. The reports may be retrieved and processed manually by an administrator of the network **120** or automatically by the manager module **138**, which may run several statistical algorithms in order to monitor the security of the network. The process may point a careless user that may connect or try to connect certain devices toward certain communication ports/buses and perform certain actions that the combination of the location and/or device and/or the action infringes the policy. The report may point a negative trend, for example, that in more than one client computer the security agent has been tampered, etc. When a portable client is not connected to the network **120** the events may be kept by the client agent and be sent to the security server **130** when the client is reconnected to the network **120**.

[0022] Policy database **135** is a database that includes a plurality of policies that may be used by the organization that owns the networks **120**. A security policy may include a set of rules that are used to determine whether a given client can be permitted to gain access to a specific device. The policy may be depended on the location of the client, on the external devices, on the type of applications, etc. A certain client may have different policies for different locations. Different policies may be allocated to different users; group of users; working hours, etc.

[0023] Client database **136** is a database that may include information regarding the various clients **110a-c** that may be connected over network **120**. Information such as but not limited to: client level of security, the type of equipment that the client possess, the external devices to which the client is allowed to be connected, information about the different environments in which the client may work, etc.

[0024] Manager module (MM) **138** manages the operation of the security server **130**. It may initiate tasks to check the situation of the security agent and the policies, which are installed in the client computers. MM **138** may create and send the appropriate policies to each one of the clients. Based on the information that is stored in the policy database **135** and the client database **136**, MM **138** may create one or more policies to a particular client. For example, a client that has a portable computer may need three policies. One policy may be used while the particular client is connected to network **120**. Another policy may be used when the client is working in a known environment, such as but not limited to his home. The last policy may be used when the client is working in an unknown location. MM **138** may run statistical algorithms over the information that is stored in the event logger **134** and may send indication and conclusions to the administrator of the network. MM **138** may receive decision regarding certain activities of a client computer and affect his connection to the network.

[0025] MMI module **139** may be a graphical user interface (GUI) that may be used by the administrator of the system to communicate with the security server **130**. The MMI may allow simple access to change policies, retrieve and check reports, update the client database **136**, etc.

[0026] FIG. 2 is a block diagram with the relevant elements of a client system **200** that may be used in an exemplary client computer **110a-c** (FIG. 1). The client system **200** may comprise one or more application programs **210a-c**, one or more device drivers **220a-c**, a security agent module **230**, one or more communication port or bus drivers (stack) **240a-c**, a core kernel module **260** and one or more physical communication ports or buses **250a-c**. Generally, the data transportation between a client computer and a device flows from/to an application **210a-c** to/from a communication port **250a-c** via the appropriate device driver **220a-c**, security agent **230** and the appropriate port driver **240a-c**. Three examples of application programs **210a-c**, device drivers **220a-c**, port drivers **240a-c** and communication ports **250a-c** are shown in Fig. 2 by way of example, and any number other than three may be used with the present invention. Client system **200** may be stored in a fix storage (e.g. a disc, flash memory, a read-only memory (ROM) etc.). During the operation of the client computer one or more of the software modules may be retrieved from the fix storage and may be loaded into a temporary memory such as a random-access memory (RAM).

[0027] The core kernel **260**, the device drivers **220a-c** and the port/buses drivers **240a-c** may be referred as the operating system (OS) of the client computer **110a-c** (FIG. 1). The OS may manage low-level aspects of computer operation, including managing execution of processes, memory allocation, file input and output (I/O), and device I/O. Exemplary OS may be windows NT or XP, Unix, MAC OS, MVS; LINUX, etc. One or more application programs **210a-c** may be transferred from a fixed storage into

the RAM for execution by the system **200**. Application program **210a-c** may be program such as but not limited to synchronization applications for PDA, Java applications for synchronization with external Java devices, such as but not limited to cellular phone, backup storage applications and communication applications such as but not limited to application that utilize Bluetooth or WiFi protocols, Internet browser, etc.

[0028] When the core kernel **260** and/or one or more application program **210a-c** may want to communicate with an external device the appropriate device drive **220a-c** may be invoked. The device driver **220a-c** is used as an intermediate between the OS **260** and/or one or more application program **210a-c** and the device itself. Exemplary external devices may be: a removable storage device, a printer, a PDA, a WiFi dongle, etc. Usually a device driver **220a-c** is supplied by the vendor of the device itself. In addition to the device driver **220a-c** a port driver **240a-c** may be invoked too. The port driver/bus driver **240a-c** is used in order to organize the communication according to the protocol that is used over the physical communication port **250a-c**. For example, if communication port **250** is a USB port than a USB driver (USB stack) is needed. The above-described computer software is for illustrating the basic desktop and server computer components that may be employed by a computer client **110a-c** (FIG. 1). In addition to those elements a security agent **230** is added by an exemplary embodiment of the present invention.

[0029] The security agent **230** may be installed in the standard storage of the client system **200** and it may be invoked during the power on cycle of the computer and remain active for the entire operation of the system. In other embodiments of the present invention the security agent **230** may be burned onto a physical memory (e.g. PROM) BIOS, etc. The security agent may be installed as a section of the OS and can be handled by an administrator having the appropriate permissions. The security agent **230** may

be installed in between the core kernel **260** and the one or more communication port/bus drivers **240a-c**.

[0030] The security agent **230** may emulate a kernel device driver and will receive the communication between the device driver **220a-c** and the core kernel **260**. During the installation and/or periodically, from time to time, the security agent **230** may register in the appropriate location in the core kernel as the first device driver for receiving the communication from/to the different communication port/bus drivers. For example, if the OS is a Microsoft product, than the security agent **230** may register in the Registry as the first device driver to get the communication. The registration may be done in a class level or in a device level. Exemplary classes may be USB, CD-ROM drivers, Hard Disk Controller, etc.

[0031] The security agent **230** may emulate a filter procedure but instead of providing the functionality of a common lower storage filter driver, the security agent performs security checking. Common operation of a lower filter is to perform device-specific functionality that is not provided by a system provider storage class device driver. The security agent **230** may emulate more than one type of lower filter driver. The number of types of lower filters that may be emulated by the security agent **230** can be configured according to the number of communication ports and devices that their transportation may be checked by the security agent **230**.

[0032] Security agent **230** may be activated when a communication port is requested. The security agent may pull the transportation to and from the communication port, processes the information and may reach a decision regarding the legality of the requested connection and/or data transfer. Security agent **230** may act as a proxy for both sides. The security agent **230** may be transparent to the user; it may not have

any icon or indication to indicate it's existence to the user. More information about the operation of security agent **230** is disclosed below in conjunction to FIG. 3, 4 & 5.

[0033] Another exemplary embodiment of the present invention (not shown in the drawings) may be used by a private user, who is not connected to a private network. The user may wish to protect the information that is stored in his computer from being copied by others. In such embodiment the client system may comprise some additional modules, which were disclosed above in conjunction with the security server **130** (FIG. 1). The additional modules may be a limited policies database **135** (FIG. 1), an event logger **134**, a manager **138** and a MMI module **139**.

[0034] FIG. 3 is a block diagram with the relevant elements of a software program **300** that may be used by an exemplary security agent **230** (FIG. 2). The software program **300** may comprise a transportation manager (TM) **310**, a buffer controller (BC) **312**, an output controller (OC) **316**, an environment detector (ED) **330**, a self checker **333**, a registry checker **336**, a bank of security policies (BOSP) **340**, a bank of protocols (BOP) **342**, a manager and decision maker (MDM) **320**, and one or more transportation processed context (TPC) **350a-c**. Three examples of transportation processed contexts (TPC) **350a-c** are shown in Fig. 3 by way of example, and any number other than three may be used with the present invention. Each one of TPC **350a-c** may comprise a parser **352**, a re-assembler **354** and an analyzer **356**.

[0035] The data transportation to/from a communication port **250a-c** (FIG. 2) may be transferred via TM **310**, BC **312** and OC **316**. TM **310** may be a software module that manages the transportation via the security agent **230**. The TM **310** may manage a table with the relevant parameters that may be needed to maintain the communication sessions that are currently transferred via the security agent. Exemplary parameters may be the source and the destination of the transportation. Each time a data

portion is transferred to or from a communication port **250a-c** (FIG. 2) the data portion is routed to the TM **310**. The data portion may be a packet for a USB communication port or a SCSI block for a SCSI bus. It should be noted that the terms "packet", "data portion" and "SCSI block" are used interchangeably herein. Henceforth, the description of the present invention may use the term 'packet' as a representative term for any of the above group. Then the packet may be transferred as is to BC **312**, which stores the packet in an appropriate location in a buffer according to the source and the destination of the communication. Then an indication may be sent to MDM **320** informing the MDM on the new packet. TM **310** may manage transportation through one or more communication ports simultaneously. More information on the operation of TM **310** is disclosed below in conjunction with FIG. 4.

[0036] BC **312** may be a software module controls a buffer (not shown in the drawings) in which the data transportation, which is currently handled by the security agent **230** (FIG. 2), may be stored. The buffer may be organized in sections; each section may be dedicated to a certain data transfer session. The data from the buffer may be retrieved and processed by an appropriate TPC **350a-c**. MDM **320** based on the result of the process that was done by the appropriate one or more TPC **350a-k** may instruct BC **312** whether to transfer the data to the output controller **316** to be sent to one or more destinations or to delete the data that belongs to this session. More information on the operation of BC **312** is disclosed below in conjunction with FIG. 4 & 5.

[0037] OC **316** may maintain the connection of the one or more data transfer sessions that are currently transferred via the security agent **230** (FIG. 2). Based on the instructions that may be received from MDM **320**, OC **316** may block the transportation to/from the appropriate communication port **250a-c**; or may send the packets to its original destination in a way that the operation of security agent **230** is transparent; or in parallel to being sent to its destination the packets may be sent to be stored in a shadowing

device (not shown in the drawing). In order to maintain the flow of operation of the client computer, when needed OC 316 may block the transportation of the data while maintaining the connection. An indication may be sent to the user, indicating that there is a problem in the data communication and the application may continue and being terminated in a normal way. More information on the operation of OC **316** is disclosed below in conjunction with FIG. 5.

[0038] ED **330** may detect the environment to which the client **110a-c** (FIG. 1) is currently connected. A client may have a portable device, such as but not limited to a laptop, which may be carried out from the organization and may be operated in other environment than the private network **120** (FIG. 1). The client may operate in the client's home where the computer may be connected to his or her Internet service provider, printer, CDROM writer etc. Another exemplary location may be a public place such as a coffee shop, hotel, airport, in which the client may be connected to the Internet via a wireless connection using a WiFi dongle or a Bluetooth dongle. In each environment a different security policy may be applied.

[0039] Identification of the environment may be based on several criteria. For example, ED **330** may have one or more environment profiles. From time to time the profiles may be loaded by the security server **130**. Each client may have its own profiles according to the environment in which the client may work. Each profile may include several parameters that may point on the environment, in which the client computer currently operates. For example, a profile may have a list of external devices to which the client may be connected in a certain environment. External devices such as but not limited to printers, external storage devices, etc. A profile may also have information on network elements that may be connected to the client computer in a certain environment. For example, gateway address, modems, RF networks name, router address, etc. In addition the profile may

include network configuration, such as but not limited to, encryption existence and type, default packet size, etc.

[0040] The information about the environment is transferred to MDM 320. The information may indicate the location of the client or may indicate that the current location is unknown. Based on the current location a security policy may be selected from BOSP 340.

[0041] BOSP 340 manages one or more security policies that are installed from time to time by the administrator of network 120 (FIG. 1), while the client is connected over network 120. BOSP 340 may include policies that are relevant to certain locations, in which the client computer may be used. Moreover the policy may be depended on the time of operation, the type of network, type of external devices etc. An exemplary architecture for organizing the different policies in BOSP 340 may be a hierarchic architecture. The top of the hierarchy may be the location, (e.g. at work, home, at a subsidiary, unknown place, etc). The second level may be the network type and configuration (e.g. wired LAN, wireless LAN such as WiFi or Bluetooth or IR, public Internet, Intranet, encryption, etc). The 3rd level may be the type of the external device (e.g. a removable storage device, removable storage media, a PDA, a cellular phone, WiFi dongle, Bluetooth dongle, a digital camera, etc.). Other exemplary method may have additional levels or may organize the BOSP 340 in other architectures.

[0042] Each policy may comprise a plurality of rules that may control a connection of a device to the client computer and the communication session between the device and the client computer. The rules may define: the maximum volume of data that can be transferred during a certain session; the maximum time for the session; type of applications (e.g. read, write, synchronization etc.) that may be used; the type

of files (e.g. doc. pdf. eml. etc.) that may be transferred; the verification method that may be used during the session to verify that the device acts as expected.

[0043] From time to time the content of BOSP 340 may be checked and updated manually by the administrator of network 120 or automatically by security server 130 (FIG. 1). From time to time MDM 320 may check that the BOSP 340 has not been tampered by hostile code. If the 340 has been damaged the MDM 320 may prevent any data transportation to/from any external device. More information on the operation of BOSP 340 is disclosed below in conjunction with FIG. 5.

[0044] BOP 342 may comprise information that may be used for parsing the packets and reassembling the content of the data that is transferred during a certain session. In addition the BOP 342 may include rules for analyzing the content of the reassembled data. The information in BOP 342 may be organized in a hierarchic architecture. The first level of the hierarchy may be associated with the type of the data communication port or bus250a-c (FIG. 2), (e.g. USB, FireWire, PCMCIA, SCSI, Infrared, wireless communication such as but not limited WiFi, Bluetooth, iSCSI, Cellular, Infiniband, Serial, Parallel, LAN port, Fiber Channel, etc). The second level may be associated with the type of the external device (e.g. a removable storage device, removable storage media, a PDA, a cellular phone, WiFi dongle, Bluetooth dongle, a digital camera, etc.). The 3rd level of BOP 342 may be associated with the application that is currently used in the communication session. For example, synchronization, data storage or backup and communication applications. Other exemplary BOP 342 may have other number of levels or may organize in other type of architectures.

[0045] For example, in case that a DiskOnKey is connected over a USB port the first level of BOP 342 may refer to a USB communication port. The USB entry may include information regarding parsing and reassembling the data

that is associated with the physical layer of the communication over a USB port. The result of processing the packet according to the information that is stored in the first level of BOP **342** may be the type of the device that is currently connected over the USB port (e.g. Digital camera, a DiskOnKey, WiFi dongle, Bluetooth dongle, etc.); the vendor ID; product ID etc.

[0046] The information in the second level of BOP **342** may refer to the type of the external device that is connected over the USB port. For example, in case that the external device is a DiskOnKey the information in the second level of BOP **342** may include information that is required to parse and reassemble the application layers of the communication. Application such as but not limited to read, write, open, close, etc. Then the 3rd level includes information regarding the application itself.

[0047] For another example, when a WiFi dongle is connected over a USB port more level of protocols are required than in the case of DiskOnKey. The WiFi dongle may allow communication over the Internet therefore six levels may be required and stored in BOP **342** in order to parse, reassemble and analyze the communication that can be transferred via a WiFi dongle. The first level of BOP **342** may refer to a USB communication port. The second level of BOP **342** may refer to a WiFi protocol such as but not limited to 802.11B, the 3rd one to Ethernet protocol, the 4th to Internet Protocol, the 5th may refer to TCP or UDP or similar protocol and the 6th may refer to the application itself that may be Microsoft Outlook, for example. More information on the operation of BOP **342** is disclosed below in conjunction with FIG. 5.

[0048] TPC **350a-c** is a temporary context that may be created by MDM **320** according to the current needs of the communication session. The first TPC **350** may be initiated after receiving a notice from the transportation manager **310** that a communication session is requested via a communication port/bus **250a-c**. Then the first context is established with parser **352**, reassembler **354** and analyzer **356** modules loaded with

the appropriate information. The information is loaded from the first level of the BOP **342** according to the type of the communication port/bus that is requested.

[0049] Additional contexts **350** may be issued during the flow of the communication by MDM **320.** Usually a context **350a-c** is associated with parsing, reassembling and analyzing a layer in the communication. Therefore when a TPC **350** collects enough information to define the next layer in the communication it may inform the MDM **320** about the next layer. The information is collected from the buffer in which the packets are stored. In response, the MDM **320** may create the next TPC **350**. MDM **320** may instruct the BOP **342** to transfer the next level of information to the new TPC **350**. The next level of information may include information on parsing, reassembling and analyzing the next communication layer that may be the device layer. At the end of the communication session the one or more TPC **350a-c** may be released. More information on the operation of TPC **350a-c** is disclosed below in conjunction with FIG. 5.

[0050] Manager **320** manages the operation of the security agent **230**. It may communicate with the security server **130** in order to download updated policies, run security tests, send reports to administrator, etc. From time to time manager **320** may receive, from ED **330**, information on the environment in which the client computer currently operates. When a communication session is initiated, manager **320** may receive an indication regarding the communication port **250** (FIG. 2) that is associated with the session. Based on the communication port, the time of the session and the information regarding the current environment, the manager may select a security policy from BOSP **340**. Then manager **320** may create a TPC **350** and instruct the BOP **342** to transfer the appropriate information that is relevant to the communication port to the new TPC **350**. When results are received from one of analyzers **356**, manager **320** may reach decisions. The decisions are based on the selected security policy. The decision may be an instruction to the output

controller **316** whether to transfer the packets from the buffers or to block the communication. A decision may be to establish additional TPC **350** in order to process the next layer. A report may be issued by manager **330**. From time to time manager **320** may request the ED **330** to initiate a learning cycle of the current environment, or manager **320** may initiate a task for checking the possibility that one or more of the modules of the security agent **230** may have been tampered.

[0051] Self-checker **333** may be invoked from time to time by MDM **320** in order to verify that the security agent has not been tampered. The time intervals between activity cycles of checker **333** may be in the range of few minutes to few hours. Checker **333** may verify that the security agent **130** is still registered in the registry as a lower filter device and the validity of the BOSP **340**. If a problem is founded MDM **320** may try to correct it, for example by register again. If it cannot be corrected the transportation via the relevant ports/buses may be blocked.

[0052] An optional module, registry checker **336**, module may be added to the security agent **230**. From time to time, while the client computer is connected over network 120 (FIG. 1), the registry checker **336** may be invoked by security server **130** (FIG. 1). The registry checker **336** may check the registry in order to verify the type of device drivers and the communication port drivers that have been active. This module may be used as a forensic tool that may deliver information to the security server **130** about the external devices that were connected to the computer client and the communication port that was used.

[0053] Another exemplary embodiment of the present invention may utilize more than one security agent modules **230**. Each module may be associated with a communication port. Other exemplary embodiments may use one or more

permanent TPC modules instead of creating a required context when it is needed. Each module may be associated with a certain protocol and/or device and/or application.

[0054] FIG. 4 illustrates a flowchart with the relevant steps of an exemplary method 400. Method 400 may be used by MDM 320 (FIG. 3) for managing input portion of data transportation via the security agent 230 (FIG. 2). Method 400 may be initiated 410 when the client computer 110a-c (FIG. 1) is turned on and may run as long as the computer 110a-c is on. Upon initializing, self-checker module 333 (FIG. 3) may be invoked. Self-checker 333 may check whether the security agent has been tampered. For example, by checking that the correct registration in the Registry is the appropriate one. The result of the self checking are sent to the MDM 320, which may use them in processing a decision of how to respond in certain data transfer sessions. At the end of the self-checker task a timer is set 413. The timer is used to define the period between repeating the self-checking process.

[0055] A decision is made 415 whether client computer 110a-c is connected over network 120 (FIG.1). If yes, then security agent 230 may initiate a communication session 418 with the security server 130 via network 120. The security agent 230 may send the result of the self-check to the server and deliver reports on the data transfer activity that has been done in the period between the last update and now. The report may include information on the files that were transferred, information on the devices that were used, the timing and the location of each data transfer session, shadowing information, etc.

[0056] During the communication session, the security server may request the security agent to perform additional tasks. For example, to invoke the registry checker module 336 in order to collect information on the different devices that have been registered between the last update and now. In addition, the server may update the bank

of the security policies 340 (FIG. 3) in the security agent. At the end of the communication session with the server 130 method 400 may proceed to step 420.

[0057] If 415 the client computer 110a-c is not connected to network 120, then method 400 may proceed to step 420 and wait for a data portion. The data portion may be a packet for a USB communication port or a SCSI block for a SCSI bus, etc. When a data portion is received, a decision is made whether 430 the data portion belongs to a new session. The decision may be based on a connection table and the relevant source or destination addresses of the data portion, or the time slot that is associated to it. The connection table may include information on the connections that are currently managed by TM 310 (FIG. 3). The information may be such as but not limited to, source and destination addresses, pointers in the buffer to the stored data that belongs to the session, port information, device information and application information, etc. The information from the connection table may be used by other modules too. Modules such as but not limited to MDM 320, one or more TPC 350a-c and OC 316.

[0058] If the session is a new session 434, then a new entry in the connection table is added, a new buffer is assigned to this session and the data portion is temporary stored in this buffer. In parallel a session task is initiated in MDM 320 in order to manage the handling of the new session. The session task may request information from the environment detector 330 and based on the current environment, and the communication port a security policy may be retrieved from BOSP 340 (FIG. 3).

[0059] Then a new TPC 350 (FIG. 3) is created to process the data. The appropriate protocol that matches the communication port is retrieved from BOP 342 (FIG. 3). Later, the internal modules of TPC 350 (Parser 352, Re-assembler 354 and Analyzer 356) are constructed with the appropriate software code to meet the specifications of the retrieved protocol. The new TPC receives the pointer to stored data and

start processing the information. In parallel, to the operation of the TPC, method 400 may proceed and continue to step 440.

[0060] In step 440 the value of timer 'T' is compared to period 'T1'. If timer 'T' is smaller than 'T1', then method 400 may return to step 420 and wait for the next data portion. If 'T' is equal or greater than 'T1', than method 400 may return to step 412 and may invoke the self-checker module again. Typical values of period 'T1" may be in the range of few minutes to few hours.

[0061] If the session 430 is not a new session, then the received data portion is stored 438 in the appropriate buffer and a pointer is stored in the connection table. The pointer is send to the appropriate TPC 350 (FIG. 3), which reviews the transportation of this session. TPC 350 may retrieve the stored data portion when it is needed to be processed.

[0062] FIG. 5 illustrates a flowchart with the relevant steps of an exemplary method 500. Method 500 may be used by MDM 320 (FIG. 3) for determining how to proceed with a session of data transportation that is currently transferred via the security agent 230 (FIG. 2). Method 500 may be initiated 510 when the client computer 110a-c (FIG. 1) is turned on and may run as long as the computer 110a-c is on. Upon initializing MDM 320 (FIG. 3) may wait 520 for receiving analyzing report from one of the analyzers 356 that are currently active. The report may be stored and a pointer may be added to the connection table. An analysis may include information on the data communication layer that is processed by the TPC 350, to which the analyzer 356 belongs. For example, if the TPC that sent the report, processes the layer of the communication port, then the report may include information on the port type, the type of the external device that is connected to the port, etc. If the level that is processed by the TPC is the device level, then the information may be on the type of application that is used. For example, if the device is a

PDA, then the reports may indicate that the current application is "synchronization application", etc.

[0063] At step **525** MDM **320** may retrieve the reports that are associated with a session. Retrieving the reports may be done by using the pointers that are stored in the connection table. The may be initiated by one or more TPC that are associated with the session, which is currently under the decision process. The reports may deliver information, such as but not limited to information on the communication port, the device, the application that is used and the type of data that is transferred. Based on the reports, the indication of the current environment and the relevant security policy, MDM **320** may reach a decision **530** how to proceed with the connection.

[0064] An exemplary security agent **230** may reach five types of decisions. The decisions may be: to block **532** the data transportation; to enable **534** the data transportation; to create **536** additional TPC; to wait **538** for additional data portion or to end **540** the communication session.

[0065] An exemplary session that may be blocked **532** by the security agent may be a communication session in which the environmental detector can not identify the environment; the report from the TPC **350**, which analyzes the port level, indicates that the communication port is USB and the device is WiFi dongle and the report from the TPC, which analyzes the device level, indicates that the application is an Email application. Another exemplary session that may be blocked **532** by the security agent may be a communication session in which the environmental detector points that the client computer **110** (FIG. 1) is connected to network **120**; the report from the TPC **350**, which analyzes the port level, indicates that the communication port is SCSI and the device is removable disc driver; the report from the TPC, which analyzes the device level, indicates that the application is "Write to Disc".

[0066] Upon receiving a decision to block 532 the data transfer of the current session, an instruction is sent to OC 316 instructing it to maintain the connection without transferring the information. For example, OC 316 may send indication to the destination that requested information is not found. In addition, information about the session may be stored in a report that may be sent to the security server 130. The information may be about the content, the time, the deriver and the application that were used, the location, etc. The resources of the security agent that have been allocated to this session may remain active in order to monitor other communication portions in the continuation of the session. For example, the application may be changed from copy information to synchronize a PDA. The new application (synchronization) may be allowed. Then MDM 320 may return to step 520 and wait to the next analysis.

[0067] An exemplary session, which may be open (allowed) 534 by the security agent 230, may be a communication session in which the report from the TPC 350, which analyzes the port level, indicates that the communication port is USB, the device is a flash memory device, such as but not limited to DiskOnKey and the report from the TPC, which analyzes the device level, discloses identification parameters of the DiskOnKey. The session may be allowed if a DiskOnKey with the same identification parameters is allowed by the appropriate security policy.

[0068] Upon receiving a decision to open 534 the data transfer of the current session, an instruction is sent to OC 316 instructing it to retrieve the appropriate data portions from the appropriate location in the buffer and transfer them toward their destination. Information about the appropriate location in the buffer and the destination may be found in the connection table. In parallel to sending the data to its destination a copy of the data may be stored for shadowing. The instruction for shadowing may be written in the policy that is used. Shadowing may be stored in a location in the disc that cannot be accessed

by the user. Indication about this session may be stored in a report that may be sent to the security server **130**. The resources of the security agent that have been allocated to this session may remain active in order to monitor changes in the session. Then MDM **320** may return to step **520** and wait to the next analysis.

[0069] An exemplary decision for initiating **536** additional TPC may be reached when a report from the TPC **350**, which analyzes the port level, indicates that the communication port is a USB port and that the device is a Bluetooth dongle, for example. Then a decision may be determined to initiate additional TPC for processing the Bluetooth section of the data communication. A security policy that matches the Bluetooth device and the current environment may be loaded to MDM **320**. The appropriate protocol may be loaded to the new parser **352** (FIG. 3) and re-assembler **354**. Information about analyzing the Bluetooth information is loaded to analyzer **356**. The previous TPC (the port TPC) may be instructed to transfer the appropriate section of the data to the new TPC for additional processing. Then MDM **320** may return to step **520** and waits to the next analysis.

In some cases additional information **538** is needed in order to reach a decision, then the MDM **320** may return to step **520** and wait for additional analysis. For example, in case that the application is "Write" to a DiskOnKey and the security policy requires checking of "Water Marks" in the content of the fill. Then the MDM may wait until the entire content of the file is been analyzed. "Water Marks" is an "undetectable" digital image with an 8 bit gray scale. The watermark is capable of carrying such information as authentication or authorization codes, or even a legend essential for image interpretation. This capability is envisaged to find application in image tagging, copyright enforcement, counterfeit protection, and controlled access to image data.

[0070] If the report that is received from the TPC, which processes the port level information, indicates that the communication session has been terminated **540**, then MDM **320** may release the resources that have been associated with the session. The resources may be like but not limited the buffers and the one or more TPCs that have been associated with the session, etc. Then MDM **320** may return to step **520** and wait to the next analysis from another data communication session.

[0071] In this application the words "unit" and "module" are used interchangeably. Anything designated as a unit or module may be a stand-alone unit or a specialized module. A unit or a module may be modular or have modular aspects allowing it to be easily removed and replaced with another similar unit or module. Each unit or module may be any one of, or any combination of, software, hardware, and/or firmware

[0072] In the description and claims of the present application the word WiFi is used to represent all types of Wireless LANs and not only 802.11b networks (for example it represent among others 802.11g, 802.11a, 802.16 etc).

[0073] In the description and claims of the present application, the word computer or client computer represent any end device, which has computing power. It includes among others cellular phones, PDAs, and other types of end equipment with a CPU that controls its behavior and communication.

[0074] In the description and claims of the present application, each of the verbs, "comprise" "include" and "have", and conjugates thereof, are used to indicate that the object or objects of the verb are not necessarily a complete listing of members, components, elements, or parts of the subject or subjects of the verb.

[0075] The present invention has been described using detailed descriptions of embodiments thereof that are provided by way of example and are not

intended to limit the scope of the invention. The described embodiments comprise different features, not all of which are required in all embodiments of the invention. Some embodiments of the present invention utilize only some of the features or possible combinations of the features. Variations of embodiments of the present invention that are described and embodiments of the present invention comprising different combinations of features noted in the described embodiments will occur to persons of the art. The scope of the invention is limited only by the following claims.

What is claimed is:

1. A method for protecting transferring of data from or to a computer, the method comprising of:

   a. getting a data portion, in a data communication session, that is transferred toward or from a communication port of the computer;

   b. storing the data portion in a temporary buffer, wherein the temporary buffer is associated with the data communication session;

   c. processing the data portion according to a protocol that is associated with the communication port;

   d. determining whether a decision on the data communication session may be reached, if not return to step 'a' and wait for the next data portion, if yes, proceed to step 'e';

   e. determining whether to allow the data communication session, if yes transferring the one or more data portions that are stored in the associated buffer toward or from the communication port, if not block the data transportation.

2. The method of Claim 1, wherein the communication port is selected from a group consisting of USB port, SCSI bus, Serial, Parallel, FireWire, PCMCIA bus, cellular, fiber channel, Bluetooth, iSCSI, wireless, LAN port, Infiniband, and Infrared.

3. The method of claim 1, wherein processing the data portion further comprising:

   (i) determining whether additional processing based on a higher level protocol is required, if not move to step 'd', if yes proceed to (ii); and

   (ii) processing part of the data portion that is relevant to the higher level protocol according to the higher level protocol and returning to step (i).

4. The method of Claim 3, wherein the higher level protocol is associated with a device selected from a group consisting of flash memory, removable hard disk drive, floppy disk, writable CD ROM, a PDA, a cellular phone, a WiFi dongle and a Bluetooth dongle.

5. The method of Claim 3, wherein the higher level protocol is associated with an application selected from a group consisting of synchronization applications for PDA, Java applications for synchronization with cellular phone, backup storage applications, Bluetooth and WiFi protocols.

6. The method of Claim 1, wherein the data portion is selected from a group consisting of packet and SCSI block.

7. The method of Claim 1, wherein getting the data portion is done by emulating a class driver.

8. The method of Claim 1, wherein getting the data portion is done by emulating a lower filter module.

9. The method of Claim 1, wherein processing the data portion according to a protocol that is associated with the communication port further comprising:

   i. parsing the data portion;

   ii. reassembling the data; and

   iii. and analyzing the reassembled data.

10. The method of Claim 1, wherein determining whether to allow the communication session is according to a security policy.

11. The method of Claim 1, wherein determining whether to allow the communication session is depending on the working environment in which the client is operating.

Fig. 1

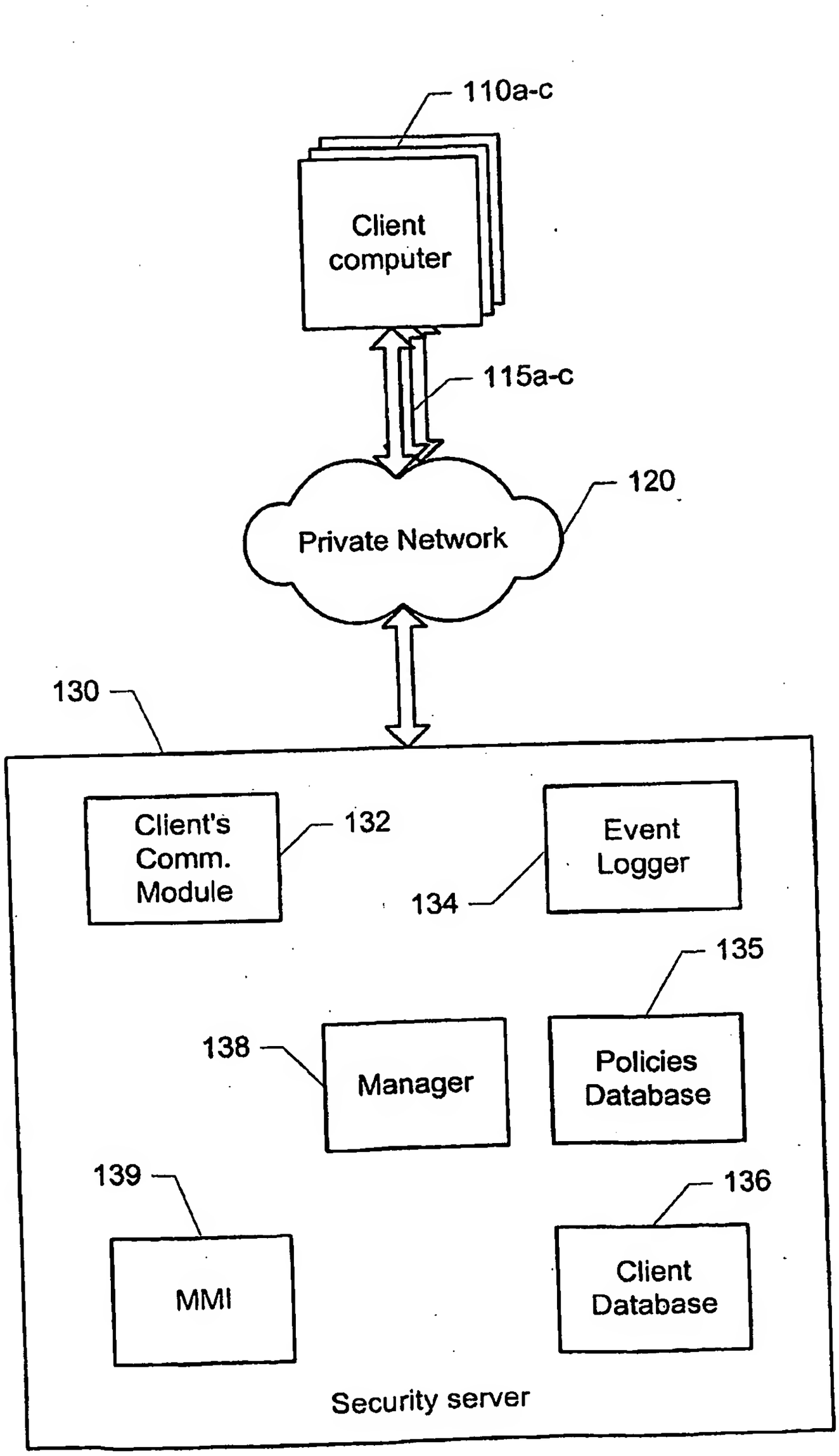


Fig. 2

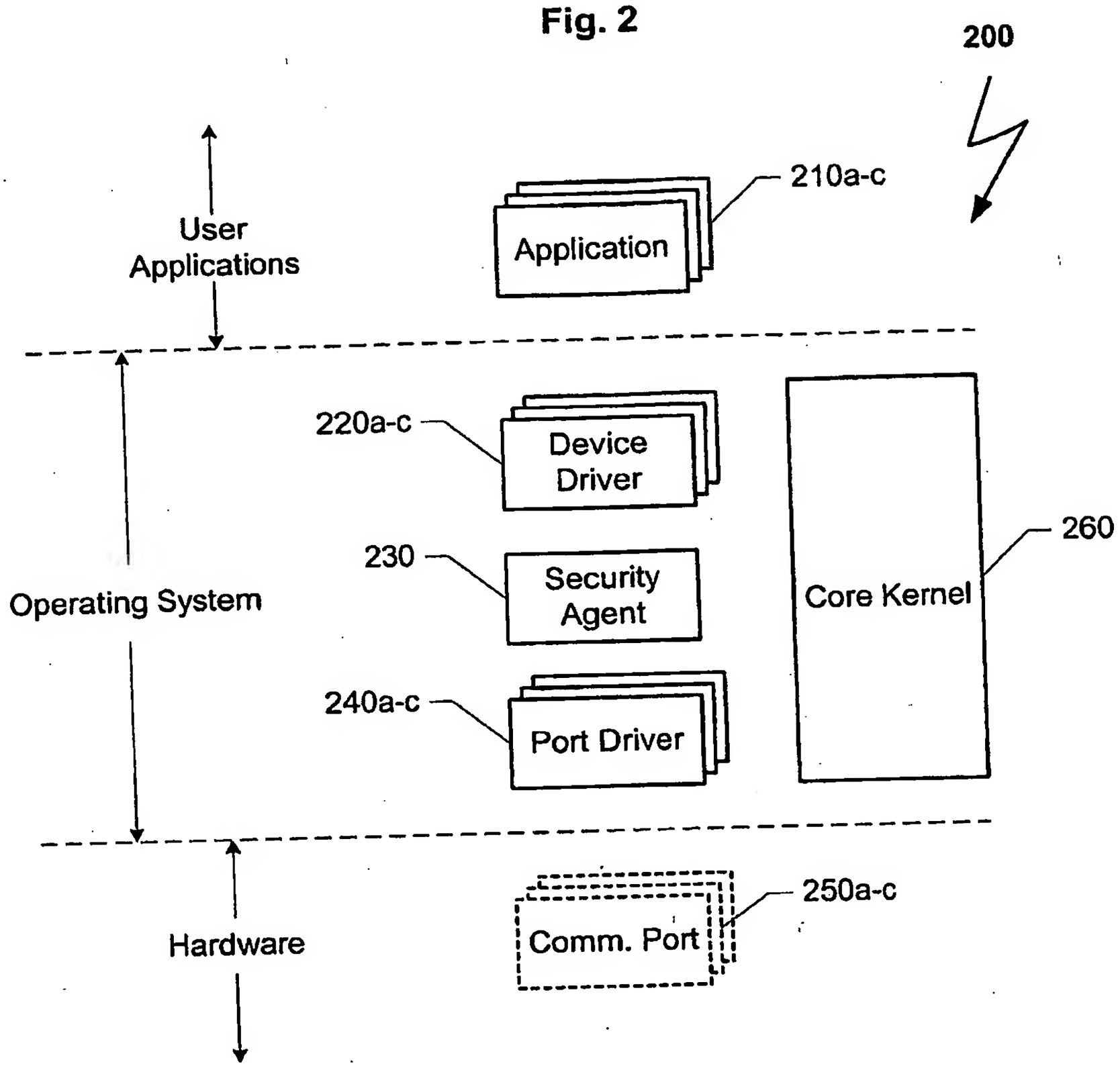
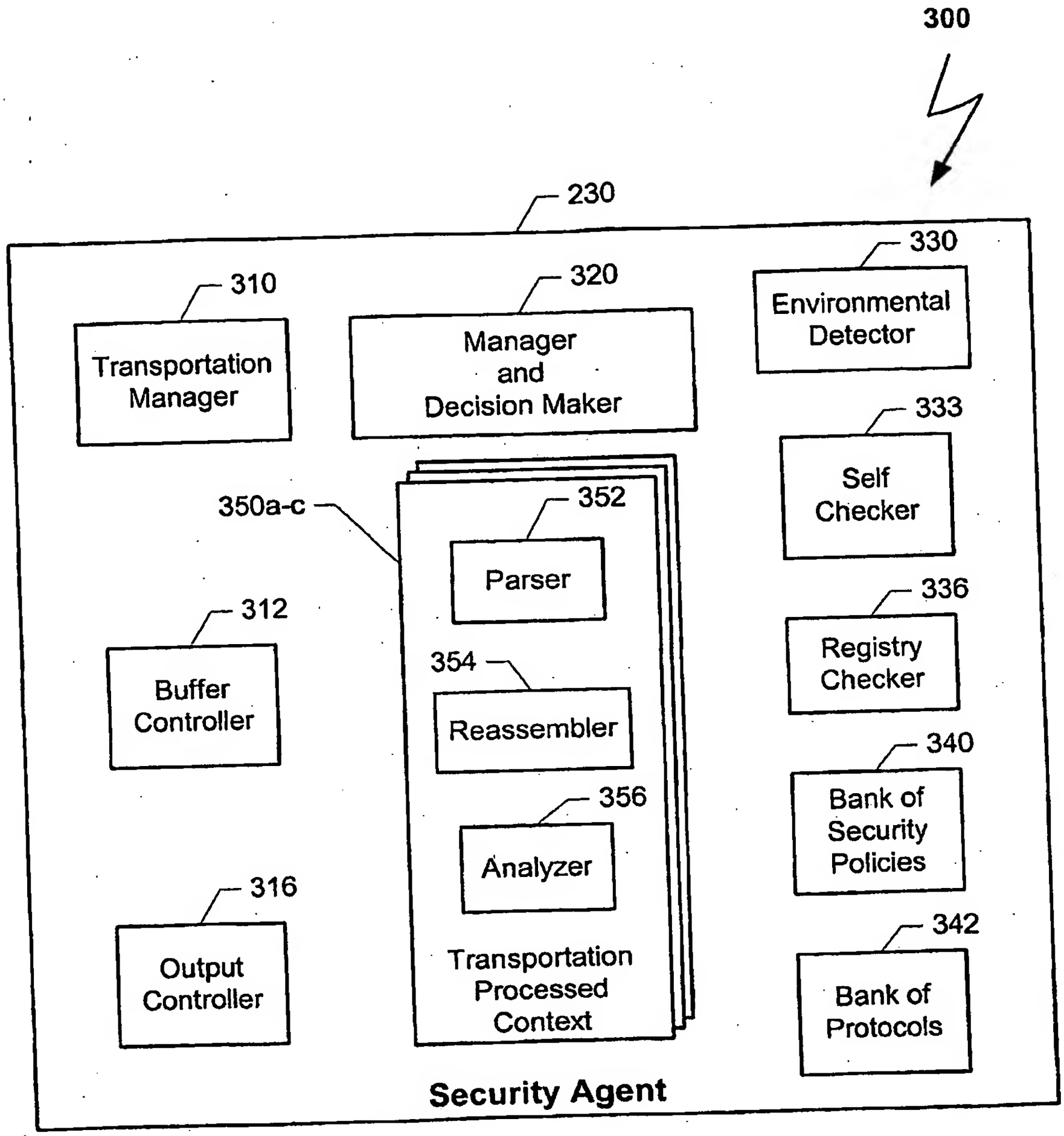


Fig. 3



-01-P

Fig. 4

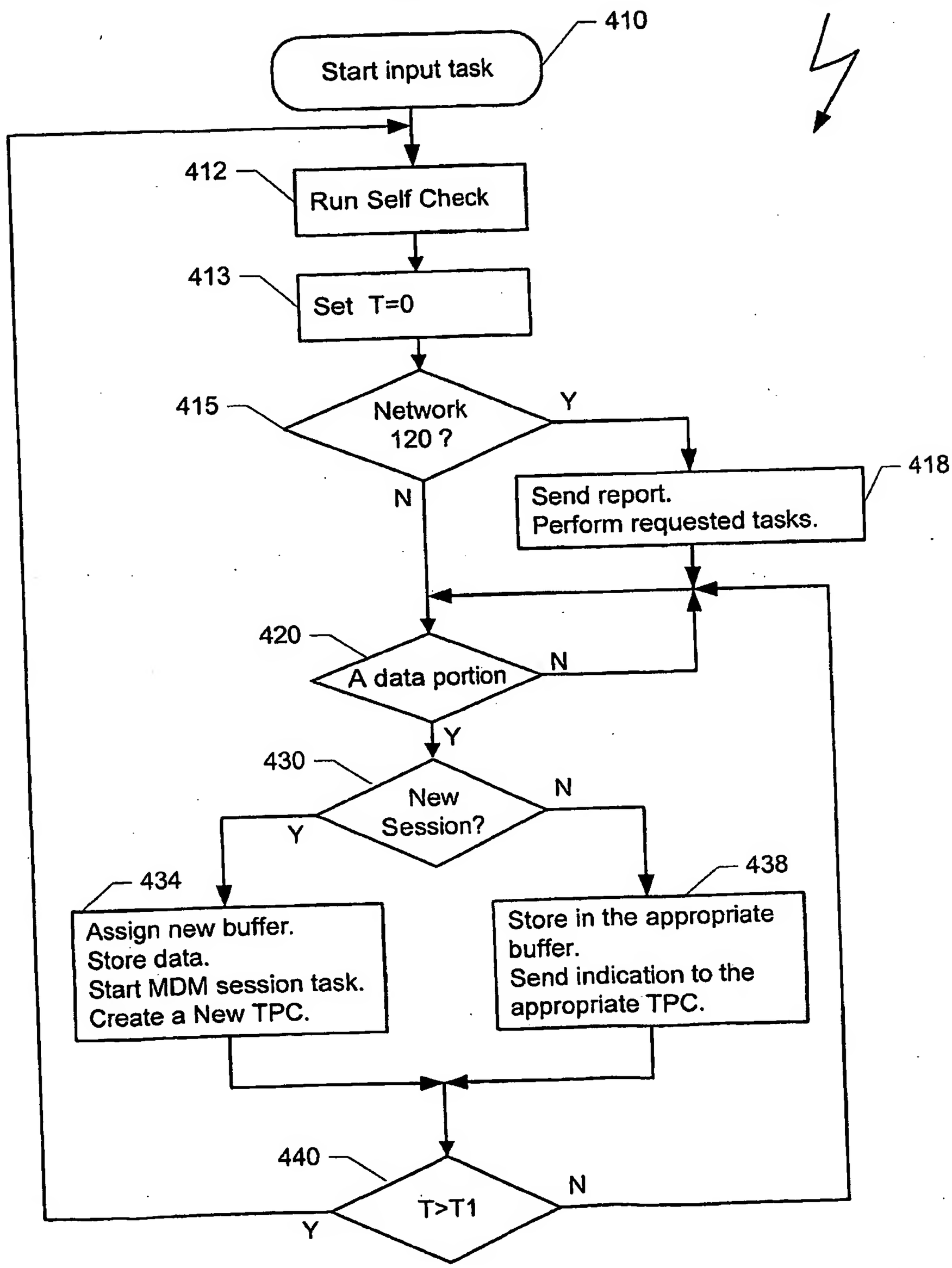


Fig. 5